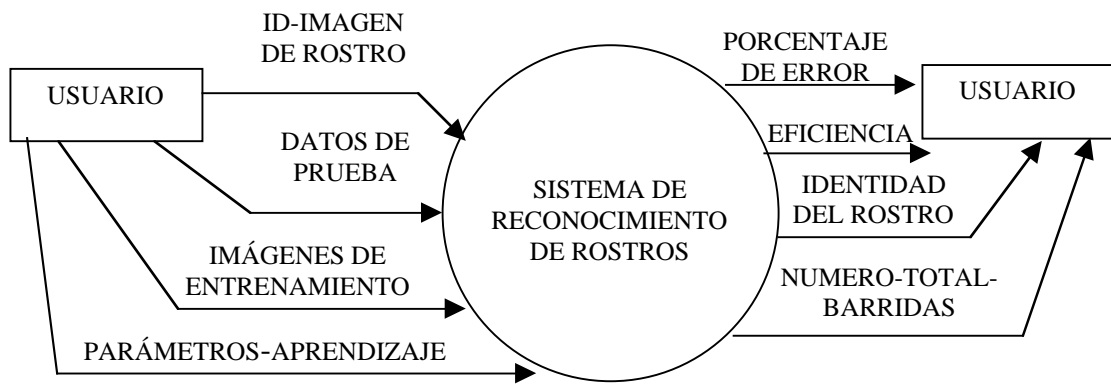


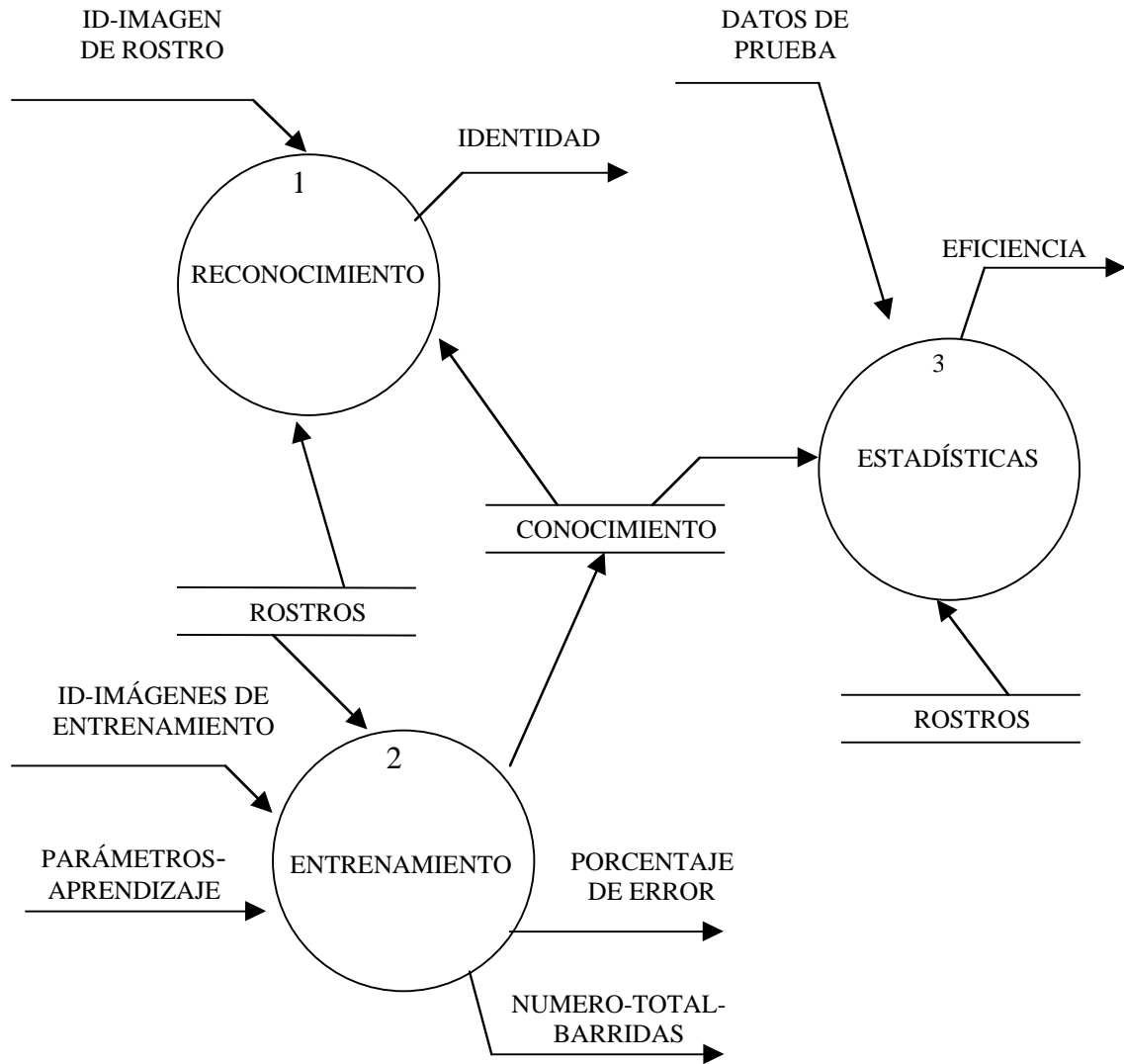
APÉNDICE A

DFD's del Sistema de Reconocimiento de Rostros

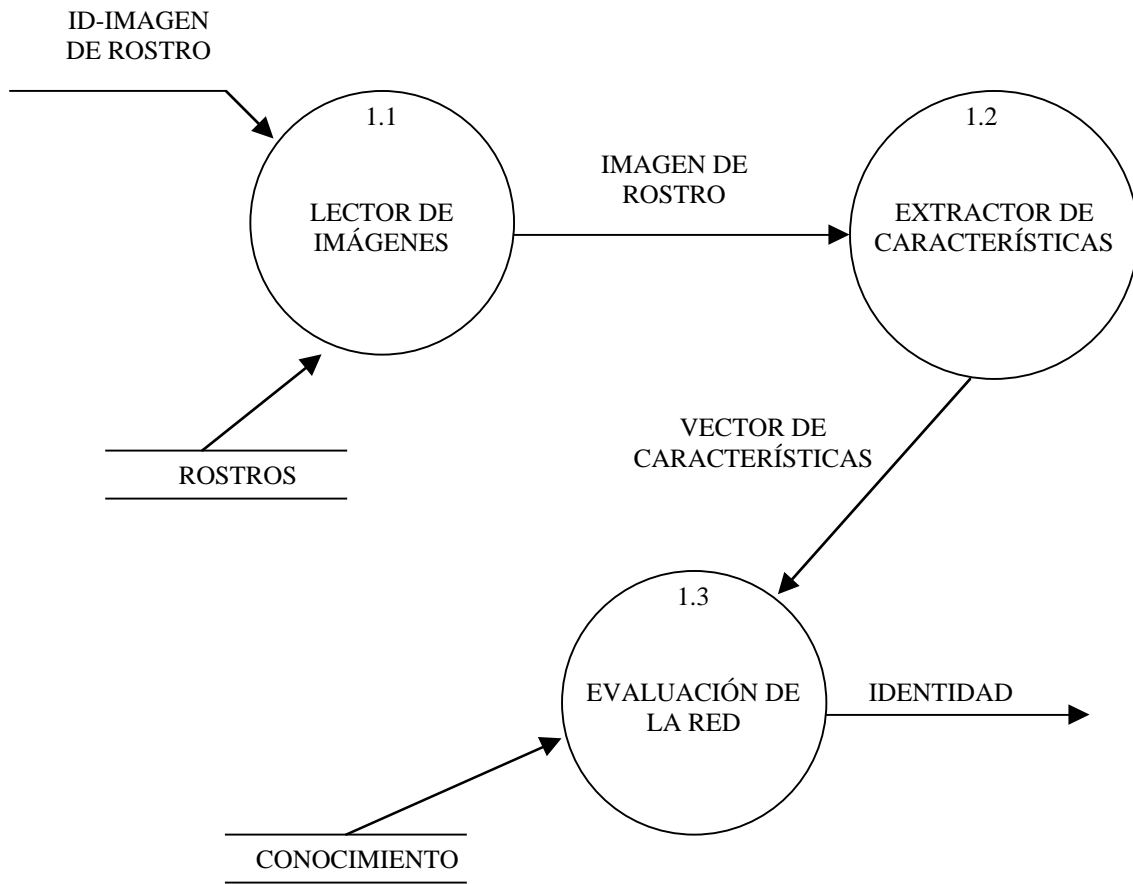
DIAGRAMA DE CONTEXTO



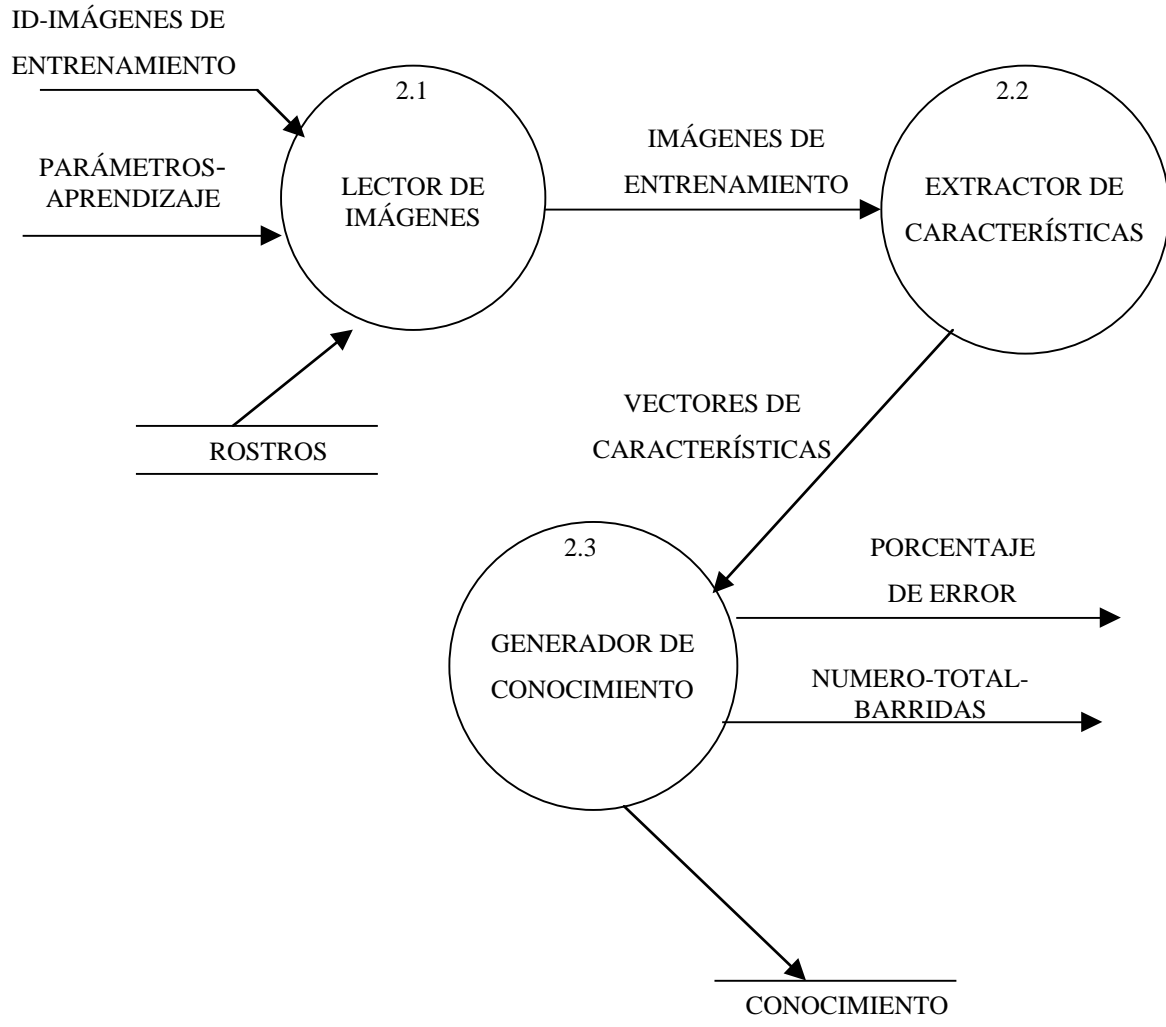
SISTEMA DE RECONOCIMIENTO DE ROSTROS



1. RECONOCIMIENTO



2 ENTRENAMIENTO



Mini especificaciones

1.1 Lector de Imágenes

LeeImagen (ID_IMAGEN_ROSTRO)

```
{  
  IF (ID_IMAGEN_ROSTRO Existe) THEN  
    Imagen_Rostro = Leer ID_IMAGEN_ROSTRO de la base de datos de ROSTROS  
  ELSE  
    RETURN Error de Lectura  
  
  RETURN Imagen_Rostro  
}
```

1.2 Extractor de Características

ExtraeCaracteristicas()

```
{  
  k = 0  
  FOR i = 0 TO Num_Ventanas_Verticales DO  
    FOR j = 0 TO Num_Ventanas_Horizontales DO  
      {  
        VectorCaracterísticas[k] = Promedio_Elementos_Ventana  
        k = k + 1  
      }  
  RETURN VectorCaracterísticas  
}
```

1.3 Evaluación de la Red

EvaluaRed()

```
{  
  Inicializar los Pesos y los Umbrales de la Red con el CONOCIMIENTO  
  FOR i = 0 TO Num_Capas_Ocultas DO  
    Calcular la salida de la capa Oculta i  
  Calcular la capa de salida
```

Calcular la minima distancia entre la salida de la red y todas las salidas deseadas como sigue:

```
    Min_Dist_Temp = MinimaDistancia(Salida_Red , Salida_Deseada_Clase[0])
Clase = 0
FOR i = 1 TO NumClases DO
{
    Min_Dist = MinimaDistancia(Salida_Red , Salida_Deseada_Clase[i])
    IF ( Min_Dist < Min_Dist_Temp )
    {
        Min_Dist_Temp = Min_Dist
        Clase = i
    }
}
Identidad = Clase
RETURN Identidad
}
```

2.1 Lector de Imágenes

```
LeeImagenes (ID_IMAGENES_ENTRENAMIENTO)
{
    FOR i = 0 TO Num_Imagenes_Entrenamiento DO
    {
        IF (ID_IMAGEN_ROSTRO Existe) THEN
            Imagenes_Entrenamiento[i] = Leer ID_IMAGEN_ROSTRO de la base de datos
        ELSE
            RETURN Error de Lectura
    }
}
RETURN Imagenes_Entrenamiento
}
```

2.2 Extractor de Características

```
ExtraeCaracteristicas(Imagenes_Entrenamiento)
{
  FOR m = 0 TO NumImagenesEntrenamiento DO
  {
    k = 0
    FOR i = 0 TO Num_Ventanas_Verticales DO
      FOR j = 0 TO Num_Ventanas_Horizontales DO
      {
        VectorCaracterísticas[k] = Promedio_Elementos_Ventana
        k = k + 1
      }
      VectoresCaracterísticas[m] = VectorCaracterísticas
    }
  }
  RETURN VectoresCaracterísticas
}
```

2.3 Generador de Conocimiento

```
GeneraConocimiento()
{
  Inicialice los pesos y los umbrales con valores aleatorios pequeños
  DO
  {
    ErrorTotal = 0
    FOR i = 0 TO Tamaño_Conj._Entrenamiento DO
    {
      Alimente a la red con el vector de Características i
      Calcule la Salida de la Red
      Calcule el Error entre la salida de la red y la salida deseada
      ErrorTotal = ErrorTotal + Error
      Ajuste los pesos de la red utilizando la regla delta generalizada
    }
  }
}
```

```
    }  
  }WHILE (ErrorTotal sea lo suficientemente pequeño)  
}
```

3. Estadísticas

```
PorcentajeGeneralizacion()  
{  
  Aciertos = 0  
  FOR i = 0 TO Datos_Prueba DO  
  {  
    Evaluación de la Red  
    IF (Salida_Red = SalidaDeseada) THEN  
      Aciertos = Aciertos + 1  
    }  
  Eficiencia = (Aciertos / Tamaño_Conj_Prueba) * 100  
  
  RETURN Eficiencia  
}
```